



software modularity group

**Gregor Kiczales**

**University of British Columbia**  
**Xerox PARC**

# a few previous “big waves”

- procedural programming  
    & block structure
- static typing
- object-orientation

## and some of their key properties

- intuitive
- efficient
- profound and incremental

### → paradigms

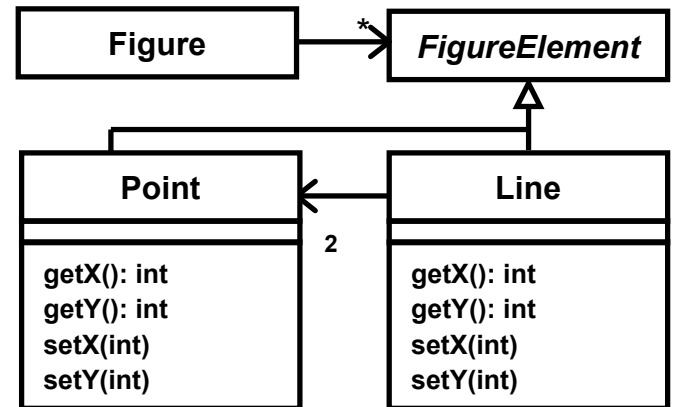
- school of: organization design, separation of concerns, abstraction, information hiding...

# object-orientation

- **model world as objects**
- **classify objects into inheritance hierarchy**

The diagram consists of three bold black words arranged horizontally: "analysis", "design", and "programming". Below these words is a long horizontal double-headed arrow pointing both left and right. Underneath the arrow, centered, is the italicized text "objects all the way through".

- **trace-ability**
- **separation of concerns**
  - within each level
  - across the levels

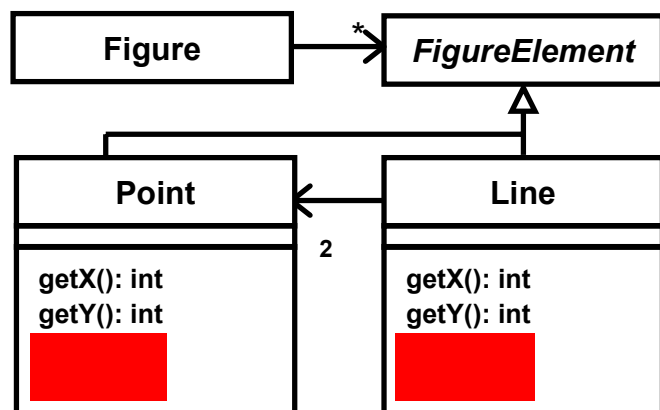


## a simple figure editor

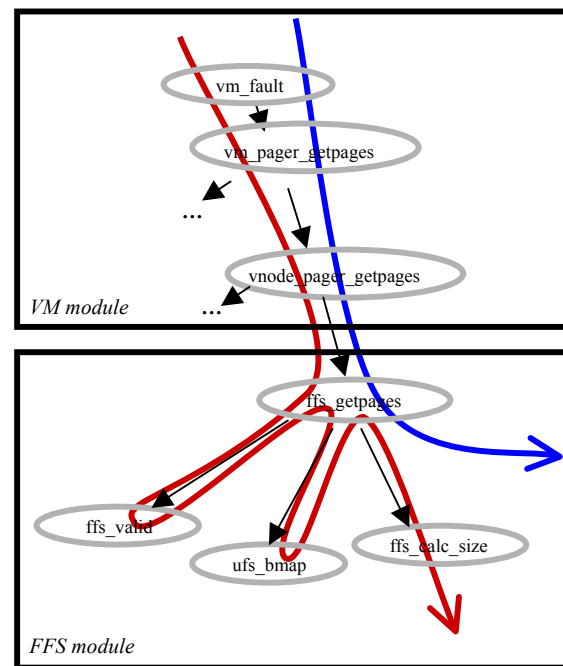
# blocks, layers & now hierarchies have limits

classic sources of complexity in embedded systems

- synchronization, access control, accounting, scheduling, performance optimization, power management, logging, context dependence...
- crosscut blocks, layers, and hierarchies



tracking when objects move in the simple figure editor



2 prefetching modes in Free BSD

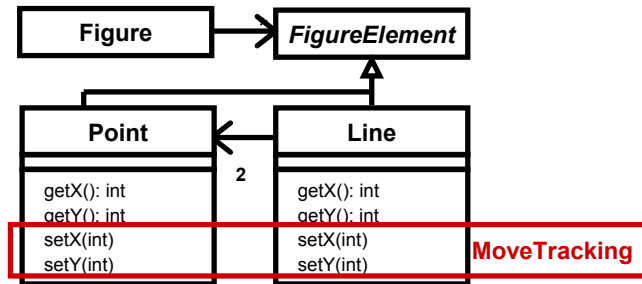
# aspects

- aspects are crosscutting units

analysis                  design                  programming

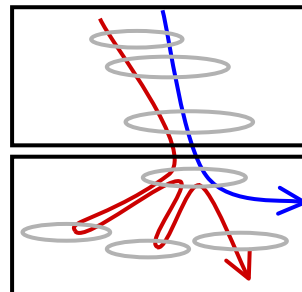
←—————→  
*aspects (& objects & procedures) all the way through*

when elements  
move



```
aspect MoveTracking {
    ... 10 loc ... }
```

detect & optimize  
sequential access



```
aspect SeqPrefetching {
    ... 10 loc ... }
```

# big steps in software development

- **not just technology**
    - languages, tools
  - **not just work practice**
    - methods, books, management
  - **synergistic combination of both**
    - intuitive, efficient, profound, incremental
    - procedural programming and related practices
    - OO programming and related practices
    - <fill this space>
- ideas that bridge the whole process